UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,344	01/02/2002	Wim De Pauw	OR920010309US2	2510

48062 7590 12/11/2009
RYAN, MASON & LEWIS, LLP
1300 POST ROAD
SUITE 205
FAIRFIELD, CT 06824

EXAMINER
MITCHELL, JASON D

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
12/11/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte WIM DE PAUW, NICK MITCHELL, MARTIN ROBILLARD,
GARY SEVITSKY, and HARINI SRINIVASAN

Appeal 2007-004513
Application 10/040,344
Technology Center 2100

Decided: December 11, 2009

Before ALLEN R. MACDONALD, *Vice Chief Administrative Patent Judge*, JOHN A. JEFFERY and STEPHEN C. SIU, *Administrative Patent Judges.*

JEFFERY, *Administrative Patent Judge.*

DECISION ON APPEAL

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1-35. We have jurisdiction under 35 U.S.C. § 6(b). We affirm and enter new grounds of rejection under 37 C.F.R. § 41.50(b).

STATEMENT OF THE CASE

Appellants invented techniques for monitoring and debugging software programs by utilizing information gathered from selected program tasks. Specifically, the user can iteratively vary the level of detail or selected program tasks of interest until the problem source is identified.[1] Claims 1, 33, and 34 are illustrative:

> 1. A method for analyzing behavior of a software system, comprising:
>
> collecting details associated with a program task associated with said software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and
>
> providing said collected details for analysis.
>
> 34. An article of manufacture for analyzing behavior of a software system, comprising:
>
> a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:
>
> a step to collect details associated with a program task associated with said software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and
>
> a step to provide said collected details for analysis.

The Examiner relies on the following as evidence of unpatentability:

Laffra	US 5,832,270	Nov. 3, 1998

[1] *See generally* Abstract; Spec. 2-3.

THE EXAMINER'S REJECTION[2]

The Examiner rejected claims 1-35 under 35 U.S.C. § 102(b) as anticipated by Laffra. Ans. 3-8.[3]

NEW REJECTION UNDER 37 C.F.R. § 41.50(B)

Although the Examiner withdrew an earlier rejection under 35 U.S.C. § 101 (Ans. 2), we nevertheless enter a new rejection under that section here in accordance with 37 C.F.R. § 41.50(b).

Claims 1, 24, 34, and 35 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. These claims are grouped as follows:

(1) claim 1 recites a method for analyzing behavior of a software system;

(2) claim 24 recites a method for tracing details associated with a program task executing in a software system;

(3) claims 34 and 35 recite articles of manufacture comprising computer readable media with code means embodied thereon for executing functions commensurate with methods (1) and (2), respectively.

FINDINGS OF FACT

The record supports the following findings of fact (FF) by a preponderance of the evidence:

[2] The Examiner withdrew a rejection under § 101. Ans. 2.

[3] Throughout this opinion, we refer to (1) the Appeal Brief filed January 11, 2007; (2) the Examiner's Answer mailed March 15, 2007; and (3) the Reply Brief filed May 15, 2007.

Appellants' Disclosure

1. Figure 1 shows an exemplary embodiment with a software analysis tool 100. The software analysis tool 100 may be embodied as a general purpose computing system, and can remotely monitor a software program 105 executing on a processor 120 remote from the processor 110 on which the software analysis tool 100 executes. Spec. 6:1-6; Fig. 1.

2. "The software analysis tool 100 includes one or more processors 110, 120 and related memory, such as a data storage device, which may be distributed or local." Spec. 6:4-6.

3. "The computer readable medium may be a recordable medium (e.g., floppy disks, hard drives, compact disks, or memory cards) or may be a transmission medium (e.g., a network comprising fiber-optics, the world-wide web, cables, or a wireless channel using time-division multiple access, code-division multiple access, or other radio-frequency channel." Spec. 9:13-17.

PRINCIPLES OF LAW

Under § 101, there are four categories of subject matter that are eligible for patent protection: (1) processes; (2) machines; (3) manufactures; and (4) compositions of matter. 35 U.S.C. § 101. While the scope of patentable subject matter encompassed by § 101 is "extremely broad" and intended to "'include anything under the sun that is made by man,'" it is by no means unlimited. *In re Comiskey*, 554 F.3d 967, 977 (Fed. Cir. 2009) (quoting *Diamond v. Chakrabarty*, 447 U.S. 303, 309 (1980)). For example, laws of nature, abstract ideas, and natural phenomena are excluded from patent protection. *Diamond v. Diehr*, 450 U.S. 175, 185 (1981).

It is the second exclusion noted above—abstract ideas—that is relevant to the appeal before us. Thus, even if the claimed invention nominally recites subject matter that falls within the enumerated categories under § 101, the claimed invention would still not recite patentable subject matter if the claim as a whole is nonetheless directed to an abstract idea. As the U.S. Supreme Court has noted, "'[a]n idea of itself is not patentable[]'....'A principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right.'" *Id.* at 185 (citations omitted).

In determining whether a claim as a whole is directed to an abstract idea, the Court has drawn a key distinction between (1) claims that seek to wholly pre-empt the use of a fundamental principle, and (2) claims that are merely limited to foreclosing others from using a particular *application* of that fundamental principle. *See In re Bilski*, 545 F.3d 943, 957 (Fed. Cir. 2008) (en banc), *cert. granted*, 77 U.S.L.W. 3442, 3653, 3656 (U.S. June 1, 2009) (No. 08-964).

Based on U.S. Supreme Court precedents, *Bilski* restated the U.S. Supreme Court's "definitive test to determine whether a process claim is tailored narrowly enough to encompass only a particular application of a fundamental principle rather than to pre-empt the principle itself." *Bilski,* 545 F.3d at 954. This restatement, embodied as the "machine-or-transformation test," requires that a claimed process either (1) be tied to a particular machine or apparatus, or (2) transform a particular article into a different state or thing. *Id.* This test ensures that the claimed process does not pre-empt uses of the principle that do not use the specified machine or apparatus. The test further precludes a claimed process from pre-empting "the use of the principle to transform any other article, to transform the same article but in a manner not covered by the claim, or to do anything other than transform the specified article." *Id.*

"[A] machine is a concrete thing, consisting of parts, or of certain devices and combination of devices. This includes every mechanical device or combination of mechanical powers and devices to perform some function and produce a certain effect or result." *In re Ferguson*, 558 F.3d 1359, 1364 (Fed. Cir. 2009) (quoting *In re Nuijten*, 500 F.3d 1346, 1355 (Fed. Cir. 2007), *reh'g denied en banc*, 515 F.3d 1361 (Fed. Cir. 2008), and *cert. denied*, 129 S. Ct. 70 (2008)).

A "manufacture" (in its verb form) is defined as the production of articles for use from raw or prepared materials by giving to these materials new forms, qualities, properties, or combinations, whether by hand-labor or by machinery. *Nuijten*, 500 F.3d at 1356 (internal quotation marks omitted). Moreover, an "article" is a particular substance or commodity. *Id.* (internal quotation marks omitted).

The Federal Circuit has recognized that the Court's precedent suggests "the 'mathematical algorithm' exception [, an example of the abstract idea judicial exception,] applies to true apparatus claims." *In re Alappat,* 33 F.3d 1526, 1542 (Fed. Cir. 1994). Thus, the mathematical exception analysis used in "[*Gottschalk* v.] *Benson*, [409 U.S. 63 (1972)] . . . applies equally whether an invention is claimed as an apparatus or process, because the form of the claim is often an exercise in drafting." *Id.* (quoting *In re Johnson*, 589 F.2d 1070, 1077 (CCPA 1978) (internal quotation marks omitted)).

If a claimed machine (or article of manufacture)[4] involves a mathematical algorithm, then we must determine whether the scope of the claimed invention encompasses one of the judicially-created exceptions. This determination of claim scope requires that we make two inquiries:

(1) Is the claim limited to a tangible practical application, in which the mathematical algorithm is applied, that results in a real-world[5] use[6] (e.g., "not a mere field-of-use label having no significance")?[7]

(2) Is the claim limited so as to not encompass substantially all practical applications of the mathematical algorithm [8] either "in all fields" of use of the algorithm or even in "only one field?"[9]

If the machine (or article of manufacture) claim involves a mathematical algorithm and fails either prong of this two-part inquiry, then the claim is not directed to patent-eligible subject matter under § 101.

[4] Notwithstanding the court's statement in *Nuijten,* 500 F.3d at 1357 n.7 ("We have never held that a *manufacture* is ever required to produce any result."), if an applicant chooses to claim the manufacture in terms of applying a mathematical algorithm (e.g., Appellants' claim 19), then this two-part inquiry applies to determine if the claim is directed to eligible subject matter under § 101.

[5]"Real-world" is not sufficient alone to establish patent-eligible subject matter absent tangibility. *See Nuijten,* 500 F.3d at 1356.

[6] *See Benson,* 409 U.S. at 68 (noting that the claim at issue was "so abstract and sweeping as to cover both known and unknown uses . . . .").

[7] *See Alappat,* 33 F.3d at 1544 (noting that the claim's recitation of "a rasterizer for creating a smooth waveform is not a mere field-of-use label having no significance.").

[8] *Benson,* 409 U.S. at 71-72.

[9] *See Bilski,* 545 F.3d at 957 (citing *Diehr*, 450 U.S. at 193 n.14) ("[I]neligibility under § 101 'cannot be circumvented by attempting to limit the use of the formula to a particular technological environment.'").

Signals are not patentable subject matter under § 101. *Nuijten*, 500 F.3d at 1355.

According to U.S. Patent & Trademark Office (USPTO) guidelines,

> A claim that covers both statutory and non-statutory embodiments . . . embraces subject matter that is not eligible for patent protection and therefore is directed to non-statutory subject matter. . . . For example, a claim to a computer readable medium that can be a compact disc or *a carrier wave* covers a non-statutory embodiment and therefore should be rejected under § 101 as being directed to non-statutory subject matter.

U.S. Patent & Trademark Office, *Interim Examination Instructions for Evaluating Subject Matter Eligibility Under 35 U.S.C. § 101*, Aug. 2009, at 2, *available at* http://www.uspto.gov/web/offices/pac/dapp/opla/2009-08-25_interim_101_instructions.pdf (emphasis in original) ("Interim Instructions").

ANALYSIS

Claims 1 and 24

Independent claim 1 recites a method for analyzing software system behavior comprising:

(1) collecting "details" associated with a program task associated with the software system based on a "specification" associated with the program task, where the specification contains one or more conditions to initiate a trace of the program task, and

(2) providing the collected details for analysis.

Similarly, independent claim 24 calls for tracing "details" associated with a program task executing in a software system comprising:

(1) monitoring the software system to identify the program task based on an associated "specification" such as that noted above, and

(2) providing "trace details" associated with the task.

Interpreting these claims as a whole, we find the recited steps are not tied to a particular machine or apparatus, nor do they transform a particular article into a different state or thing. We address each prong of the machine-or-transformation test in turn.

Claims 1 and 24 Are Not Tied To a Particular Machine Or Apparatus

Claims 1 and 24 are not tied to any machine—let alone a particular machine as *Bilski* requires. "[A] machine is a concrete thing, consisting of parts, or of certain devices and combination of devices. This includes every mechanical device or combination of mechanical powers and devices to perform some function and produce a certain effect or result." *Ferguson*, 558 F.3d at 1364 (internal quotation marks omitted).

Claims 1 and 24 are not tied to any concrete parts, devices, or combinations of devices, let alone a concrete thing that performs a function and produces a certain effect or result. Rather, claim 1 merely calls for (1) collecting "details" associated with a program task based on an associated "specification" that contains "conditions" to initiate a trace of the task, and (2) providing the collected details for analysis. Similarly, claim 24 calls for tracing "details" associated with a program task executing in a software system comprising (1) monitoring the software system to identify the program task based on an associated "specification" noted above, and (2) providing "trace details" associated with the task.

Not only are the claims silent as to what machine—if any—is used to perform these data gathering steps, nothing in claim 1 precludes *mentally* collecting and providing these "details" for analysis as claimed—a cognitive activity that is not patentable under § 101. *See Benson*, 409 U.S. at 67; *see also Comiskey*, 554 F.3d at 979 ("[M]ental processes-or processes of human thinking-standing alone are not patentable even if they have practical application.").

Moreover, the recited "specification" and its associated conditions are likewise not tied to a machine—let alone a particular machine. Although the conditions are related to initiating a trace of a program task, collecting details based on these conditions is simply not tied to a machine, let alone a particular machine, as *Bilski* requires.

Nevertheless, even assuming that a machine is required to implement the recited method (which it is not), such a machine would not be a *particular* machine as *Bilski* requires. *See Bilski*, 545 F.3d at 961-62. Appellants' Specification all but confirms this point. According to Appellants' Specification, the software analysis tool 100 may be embodied as a general purpose computing system. FF 1. This generalized computing system, however, does not tie the process to a *particular* machine or apparatus.[10] Even if claim 1 was tied to such a general purpose computer (which it is not), such a nominal recitation of physical structure tantamount to a general purpose computer would be analogous to the recitation of storing binary coded decimal signals in a shift register that the U.S. Supreme

[10] *See Benson*, 409 U.S. at 63 (holding that claims directed to a method for converting binary-coded-decimal (BCD) numerals into pure binary numerals for use with a general-purpose digital computer were nonstatutory under § 101).

Court found to be unpatentable in *Benson*. *See Benson*, 409 U.S. at 73 (listing claim 8 which calls for, in pertinent part, "storing the binary coded decimal signals in a *reentrant shift register*") (emphasis added). In any event, "[n]ominal recitations of structure in an otherwise ineligible method fail to make the method a statutory process." *Ex parte Langemyr*, App. No. 2008-1495, slip op. at 20 (BPAI May 28, 2008) (Informative) (citing *Benson*, 409 U.S. at 71-72), *available at* http://www.uspto.gov/web/offices/dcom/bpai/informative_opinions.html.

For the foregoing reasons, we find that independent claims 1 and 24 are not tied to a particular machine.

Claims 1 and 24 Do Not Transform a Particular Article Into a Different State or Thing

We also find that claims 1 and 24 do not transform a particular article into a different state or thing. Merely collecting "details" associated with a program task for a software system, and "providing" those collected details for analysis in claim 1 is tantamount to data gathering steps and is therefore insignificant extra-solution activity. *See Bilski,* 545 F.3d at 962 ("[T]he involvement of the machine or transformation in the claimed process must not merely be insignificant extra-solution activity."); *see also id.* at 963 (characterizing data gathering steps as insignificant extra-solution activity); *Parker v. Flook*, 437 U.S. 584, 588-90 (1978) (insignificant post-solution activity step found to be insufficient to impart patentability).

We reach a similar conclusion regarding claim 24 which calls for (1) monitoring the software system to identify the program task based on an associated "specification" with conditions to initiate a trace, and (2) providing "trace details" associated with the task. Here again, since these steps are merely gather data, they likewise constitute insignificant extra-solution activity. *See Bilski,* 545 F.3d at 962-63.

Although Appellants argue that the recited trace of details constitutes a transformation of the original code since it is a summary of its execution (App. Br. 5), the limitation nonetheless is tantamount to insignificant extra-solution activity. *See id.*

Bilski does recognize that certain types of data can be transformed to meet the transformation test. For example, the court cites the X-ray attenuation data in *In re Abele*, 684 F.2d 902 (CCPA 1982) which not only represented physical and tangible objects, but was also transformed into a particular visual depiction of a physical object on a display. *Bilski*, 545 F.3d at 962-63. But here, claims 1 and 24 recite no such transformation.

Lastly, Appellants' contention that this collection or trace of details "provides a useful, concrete and tangible result" (App. Br. 5) is likewise unavailing. The "useful, concrete, and tangible result" test pronounced in *State Street Bank & Trust Co. v. Signature Financial Group*, 149 F.3d 1368, 1373 (Fed. Cir. 1998) is no longer adequate. *Ferguson*, 558 F.3d at 1364 n.3 ("[T]he 'useful, concrete and tangible result test' is insufficient to determine whether a claim is patent-eligible under § 101 . . . and is inadequate." (quoting *Bilski*, 545 F.3d at 959-60 (internal quotation marks omitted)).

Therefore, the process of claims 1 and 24 does not transform a particular article into a different state or thing.

For the foregoing reasons, independent claims 1 and 24 fail to recite statutory subject matter under § 101. Although we decline to reject every claim under our discretionary authority under 37 C.F.R. 41.50(b), we emphasize that our decision does not mean the remaining claims are patentable. Rather, we merely leave the patentability determination of these claims to the Examiner. *See* MPEP § 1213.02.

Claims 34 and 35

We reach a similar conclusion regarding claims 34 and 35 which recite articles of manufacture comprising computer readable media with code means embodied thereon for executing functions commensurate with the methods of claims 1 and 24, respectively.

The Specification explains that the recited computer readable medium may not only be a recordable medium, but also a transmission medium that conveys a signal. *See* FF 3. Signals, however, are not patentable subject matter under § 101. *Nuijten*, 500 F.3d at 1355.

Thus, when read in light of the Specification, independent claims 34 and 35 include both statutory subject matter (recordable media) and non-statutory subject matter (signals conveyed by transmission media). Such claims, however, must be amended to recite solely statutory subject matter. *See* Interim Instructions, at 2.

For the foregoing reasons, claims 34 and 35 do not recite statutory subject matter under 35 U.S.C. § 101.

THE ANTICIPATION REJECTION

Regarding claim 1,[11] the Examiner finds that Laffra discloses all of the claimed subject matter. Ans. 3 and 9. Specifically, Laffra's "hooks" are said to pass information to a monitoring function, and therefore initiate a trace of a program task. Ans. 9. The Examiner adds that Laffra's visualization script provides "conditions" in the form of visualization rules to initiate a trace. *Id.*

Appellants argue that Laffra does not use conditions to initiate a trace of a program task as claimed. According to Appellants, Laffra's hooks are not conditional instructions and, as such, are executed when encountered. App. Br. 6; Reply Br. 3. Appellants add that Laffra does not disclose that the script determines or controls when or what information is generated by the hooks. *Id.*

Regarding claims 2-4 and 25, Appellants argue that Laffra does not define a program task duration by the one or more conditions associated with a state of the software system, where the conditions include entry or exit of a specified method and creating or deleting at least one specified object.

Regarding claims 6 and 8, the Examiner finds that Laffra teaches passing at least one object or scalar value (i.e., a comparison operator) that triggers the display, and that this comparison operation is used as a basis to compare a specified resource to a threshold. Ans. 5 and 10.

[11] Appellants argue the following claim groupings: (1) claims 1, 24, and 32-35; (2) claims 2-4 and 25; (3) claims 6 and 8; and (4) claim 7. *See* App. Br. 5-9. Accordingly, we treat each group separately. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Appellants argue that Laffra's assigning new values to global variables or local variables is a condition that defines a program task duration, let alone a condition with the characteristics recited in claims 6 and 8. App. Br. 8-9; Reply Br. 4-5.

Regarding claim 7, the Examiner notes that Laffra's method triggers can be used to count the number of times a given method is executed, and to update the display when a certain threshold is reached. Ans. 5 and 10. Appellants, however, argue that Laffra does not disclose that a specified sequence of method invocations is a condition that defines a duration of program task. App. Br. 9.

The issue before us, then, is as follows:

ISSUE

Under § 102, have Appellants shown that the Examiner erred by finding that Laffra:

(1) collects details associated with a program task associated with a software system based on a specification associated with the program task, where the specification contains one or more conditions to initiate a trace of the program task as recited in claim 1?

(2) defines a program task duration by the one or more conditions associated with a state of the software system, where the conditions include entry or exit of a specified method, and creating or deleting a specified object as recited in claims 2-4 and 25?

(3) discloses conditions including (a) a passing of at least one specified object or scalar value as an argument, return value, or field value,

and (b) a resource exceeding a specified threshold as recited in claims 6 and 8?

(4) discloses conditions including a specified sequence of method invocations as recited in claim 7?

FINDINGS OF FACT

The record supports the following additional findings of fact (FF) by a preponderance of the evidence:

Laffra

4. Laffra discloses a system that can automatically insert method "hooks"[12] 260, 270 into object-oriented software to visualize its execution. The hooks can generate graphical information that can be visualized on a graphical interface. When a method hook is run, it can indicate (1) creating or destroying an object instance, or (2) entering or exiting a method in the object class. Laffra, Abstract; col. 3, l. 48 – col. 4, l. 18; Fig. 2.

5. Upon one of these occurrences, the method hooks initiate execution of a monitoring function 300 that uses (1) the graphical information, and (2) a visualization script 285 with rules 288 to update a visualization shown on the graphical interface. The script 285 tells the monitoring function how to interpret the information generated by the hooks. Laffra, Abstract; col. 5, l. 37 – col. 6, l. 62; Figs. 2-3.

[12] "Hooks" are executable code (data structures) that are used to pass information to a monitoring function. Laffra, col. 3, ll. 51-52; col. 4, ll. 12-13.

6. The visualization script 285 identifies one or more object classes and specifies how to display instances of those object classes on display 126. This specification is done using the visualization script rules 288. Laffra, col. 5, ll. 44-48; Figs. 2-4.

7. Information gathered by the monitoring function is visualized on a graphics display, guided by the set of rules 288 in the visualization script 285. Each time a particular hook is executed, the monitoring function (1) inspects the current display and script, and (2) modifies the display depending on the hook and visualization script. Laffra, col. 6, ll. 1-7; Fig. 2-3.

8. Method triggers can be used to (1) change the display when a given method is entered (e.g., by coloring a visual item red), and (2) reset the display the method is left again (e.g., coloring the item back to green). Laffra, col. 7, ll. 43-48; Fig. 3.

9. The monitoring function resolves a set of constraints possibly defined by the script. The resolution of the constraint rules can result in updates to visualization variables, thereby influencing the display. Laffra, col. 7, ll. 49-53; Fig. 3 (step 370).

10. Constraints define relationships over two sets of variables, and can take the form of two operands and one operator (a comparison operator such as ‘<’, ‘>=’, ‘==”). Laffra, col. 9, ll. 2-11.

11. When the class and method names of the hook match those of the trigger, the trigger matches the hook, and the corresponding action is executed (including assigning new values to global local variables). Laffra, col. 9, ll. 17-25.

12. Laffra discloses an example where a method 230 includes two lines of code (8a and 8b) corresponding to a hook. Laffra, col. 4, ll. 40-48.

13. Laffra notes that method triggers can be used to count the number of times a given method is executed, and to update the display when a certain threshold is reached. Laffra, col. 7, ll. 40-43.

PRINCIPLES OF LAW

Anticipation is established only when a single prior art reference discloses, expressly or under the principles of inherency, each and every element of a claimed invention as well as disclosing structure which is capable of performing the recited functional limitations. *RCA Corp. v. Appl. Dig. Data Sys., Inc.*, 730 F.2d 1440, 1444 (Fed. Cir. 1984); *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1554 (Fed. Cir. 1983).

ANALYSIS

Claims 1, 24, and 32-35

Based on the record before us, we find no error in the Examiner's anticipation rejection of claims 1, 24, and 32-35. Laffra's hooks initiate execution of a monitoring function that uses information from both the hooks and the visualization script to update a display. FF 4-7. As such, the hooks, visualization script, and its associated rules collectively provide a "specification" with "conditions" to initiate a trace of the program task whose details are ultimately displayed. *See id.*

Notably, when a hook is executed, it can indicate (1) creating or destroying an object instance, or (2) entering or exiting a method. FF 4. The monitoring function then interprets this information via the script and its associated rules to identify the associated program task and update the display. FF 5-8. These functions, initiated in part by interpreting the hooks' information, reasonably constitute a specification with conditions to initiate a program task trace as claimed.

For the foregoing reasons, Appellants have not persuaded us of error in the Examiner's rejection of claims 1, 24, and 32-35. Therefore, we will sustain the Examiner's rejection of those claims.

Claims 2-4 and 25

We will also sustain the Examiner's rejection of claims 2-4 and 25. We agree with the Examiner (Ans. 9-10) that Laffra's hooks, at least in part, define a program task duration by passing information to the monitoring function each time a method is entered and exited. *See* FF 4 and 5. These conditions also include the creation or destruction of an object instance (FF 4)—conditions that would also effectively define the duration of a task associated with that object.

We also agree with the Examiner (Ans. 10) that the period of time associated with changing the display to a different color (e.g., red) in conjunction with entering and exiting methods (*see* FF 8) likewise would define a duration as claimed.

For the foregoing reasons, Appellants have not persuaded us of error in the Examiner's rejection of claims 2-4 and 25.

Claims 6 and 8

We will also sustain the Examiner's rejection of claims 6 and 8. We see no error in the Examiner's finding (Ans. 4, 5, and 10) that Laffra's hook code (FF 12) would at least involve passing an object or scalar value to trigger a display. Moreover, nothing in claims 6 and 8 precludes the monitoring function's resolving a set of constraints that involve updating variables (which include a comparison operator) to execute a corresponding action (FF 9-11). Nor does the scope of claim 8 preclude Laffra's updating the display when a certain number of method executions is reached. *See* FF 13.

For the foregoing reasons, Appellants have not persuaded us of error in the Examiner's rejection of claims 6 and 8.

Claim 7

We will also sustain the Examiner's rejection of claim 7. We see no error in the Examiner's reliance (Ans. 5 and 10) on Laffra's use of method triggers to count the number of times a given method is executed, and to update the display when a certain threshold is reached (FF 13). Notably, nothing in the claim precludes a sequence involving multiple invocations of a method. And as we indicated previously, nothing in claim 2 precludes Laffra's hooks from defining a program task duration by passing information to the monitoring function each time a method is entered and exited. *See* FF 4 and 5.

For the foregoing reasons, Appellants have not persuaded us of error in the Examiner's rejection of claim 7.

CONCLUSIONS

Appellants have not shown that the Examiner erred in rejecting claims 1-35 under § 102. We have also entered new grounds of rejection for independent claims 1, 24, 34, and 35 under § 101.

ORDER

The Examiner's decision rejecting claims 1-35 is affirmed. We have also entered new grounds of rejection for claims 1, 24, 34, and 35.

This decision contains new grounds of rejection pursuant to 37 C.F.R. § 41.50(b) that provides that "[a] new ground of rejection . . . shall not be considered final for judicial review."

Section 41.50(b) also provides that the Appellants, WITHIN TWO MONTHS FROM THE DATE OF THE DECISION, must exercise one of the following two options with respect to the new ground of rejection to avoid termination of the appeal as to the rejected claims:

> (1) Submit an appropriate amendment of the claims so rejected or new evidence relating to the claims so rejected, or both, and have the matter reconsidered by the examiner, in which event the proceeding will be remanded to the examiner. . . .
>
> (2) Request that the proceeding be reheard under § 41.52 by the Board upon the same record. . . .

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED
37 C.F.R. § 41.50(b)

pgc

RYAN, MASON & LEWIS, LLP
1300 POST ROAD
SUITE 205
FAIRFIELD CT 06824